NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

The GMPLS Lightwave Agile Switching Simulator GLASS - An Overview -

Version: Draft 1.0

ABSTRACT:

The GMPLS Lightwave Agile Switching Simulator (GLASS) is a simulation engine that allows the modeling and performance evaluation of routing, restoration, and signaling protocols for optical networks. It is designed as an extension to the Scalable Simulation Framework Network (SSFNet) and provides the user with a high degree of link and optical layer details. It contains several optical layer components such as fibers, lambdas, and optical switches. This document gives a brief overview of the GLASS simulator.

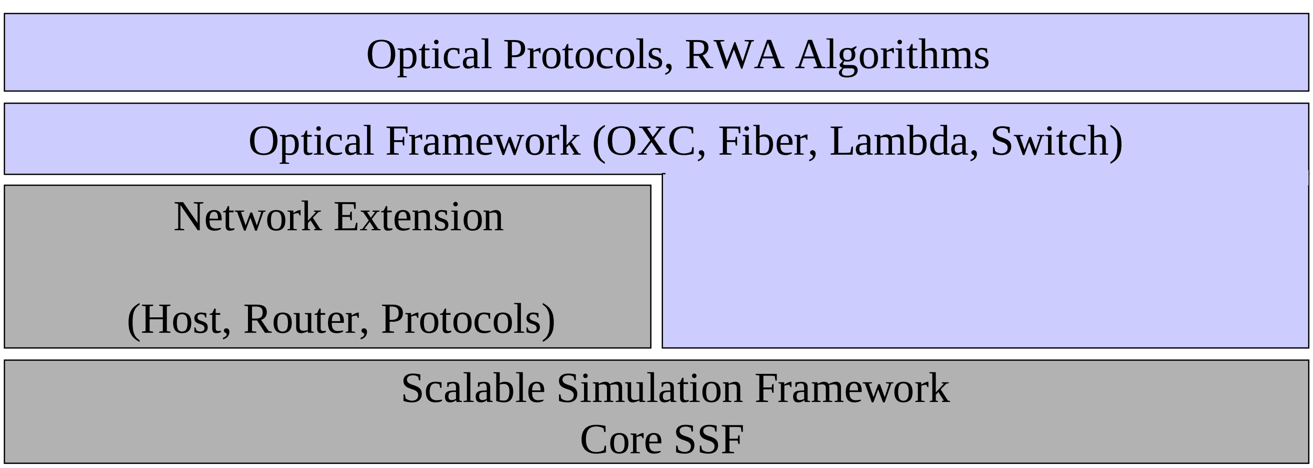
TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

AS more intelligence and control is added into optical networks (ONs), new standards such as the Generalized Multi-Label Switching (GMPLS) are emerging in order to allow edge networking products like routers and switches to dynamically interoperate with the optical layer components in order to request bandwidth, set up protection path, invoke recovery procedures. The GMPLS Lightwave Agile Switching Simulator (GLASS) is an optical simulation framework, that facilitates the evaluation of routing, restoration and signaling protocols in an optical environment and allows network planners and researcher to study behavior of algorithms and protocols without the need for building a real implementation. GLASS evolved from MERLiN [3], a simulation tool focused on the evaluation and modeling of routing and wavelength assignment algorithms in WDM networks. In 2001, GLASS was redesigned based on the Scalable Simulation Framework Network (SSFNet) [1].

We can identify at least two user groups for GLASS. There are those who are interested in the behavior analysis of GMPLS and IP-related protocols, and those who are interested in studying the optical layer without using MPLS-like protocols. A third category includes users who are interested in both the GMPLS and ON worlds.
GLASS has a modular architecture consisting of several components, namely, the optical physical layer, the logical layer, and protocols such as routing, restoration, and wavelength assignment algorithms. Some example implementations of failure recovery protocols are included in this release. The implementation allows the addition of new algorithms as well as protocol implementations.



2 THE SIMULATOR DESIGN

Glass is designed on top of SSF/SSFNet and takes advantage of the many capabilities provided by SSFNet, such as the handling of discrete events, the ability to run on multiprocessors, and the scalability to large number of nodes. Building it on top of SSFNet allowed us to focus on the optical components, their architecture and behavior. GLASS uses the Data Modeling Language (DML)[2] to design the topology and derive scripts for the simulation scenarios. DML also allows a very high level description of the components and the configuration topology.

Physical and Logical Layer Topology

The Topology consists of two layers. Layer 1 consists of the pure physical layer topology with components such as OXC, Links, Fibers, Lambdas, Switches, and other hardware components; Layer 2 contains information such as the route, the optical path, and segments in the optical path. Figure 2 shows the two layers of the topology that are provided in the framework.

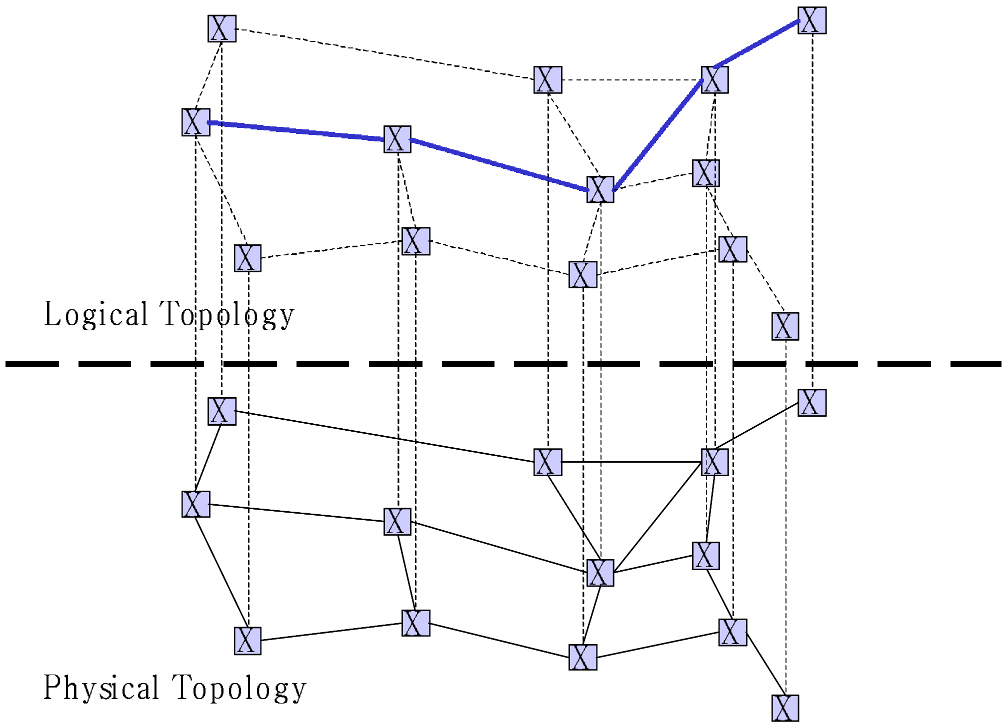


Figure 2: A Layered Topology

2.1 THE TOPOLOGY COMPONENTS

The topology provides the following basic components:

- Optical Cross Connect: OXC and OXCEdgeRouter
- Label Switched Router: LSR
- Optical Network Interface Card: ONIC
- Optical Link
- Fiber
- Lambda
- Optical Switch

Each of these components is configurable and includes a number of parameters. Figure 3 shows the design of an Optical Cross Connect (OXC).

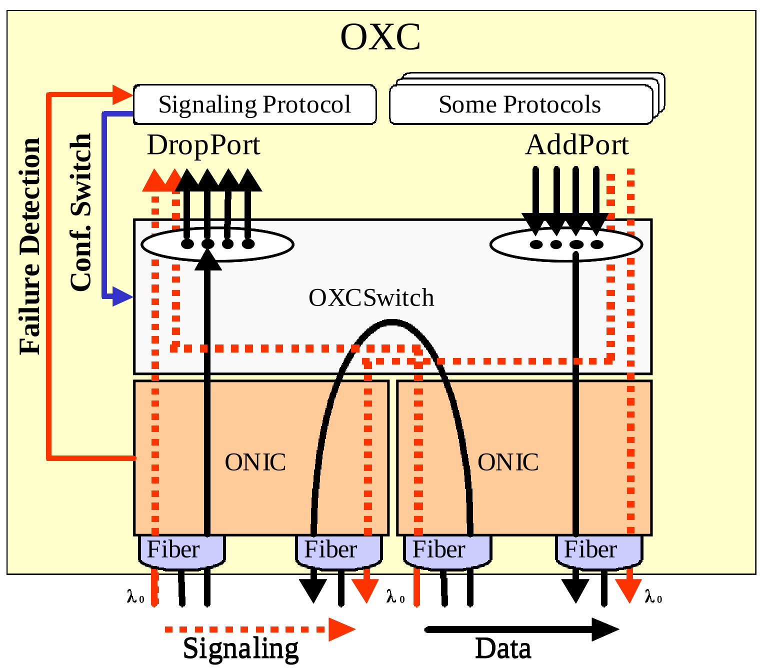


Figure 3: The Optical Cross Connect (OXC)

The OXC provides both, O/O/O switching and O/E/O switching. The O/E/O switching is implemented by using the Add/Drop capability to pass incoming messages to the upper layer and send outgoing messages via the switch to the corresponding fiber. The configuration of the switch is done externally by a signaling protocol. For designers of centralized-control algorithms, the framework provides the necessary tools to access each switch and configure it directly.

The optical link, shown in Figure 4, is a logical bundle of multiple optical fibers. An optical link can be both, unidirectional and bidirectional. Each fiber can be unidirectional and bidirectional as well. The fiber itself contains a bundle of lambdas. The lambdas are grouped in data lambdas and signaling lambdas. This is done to reserve lambdas for a special use. This also has an effect on the calculation of the bandwidth of a fiber.

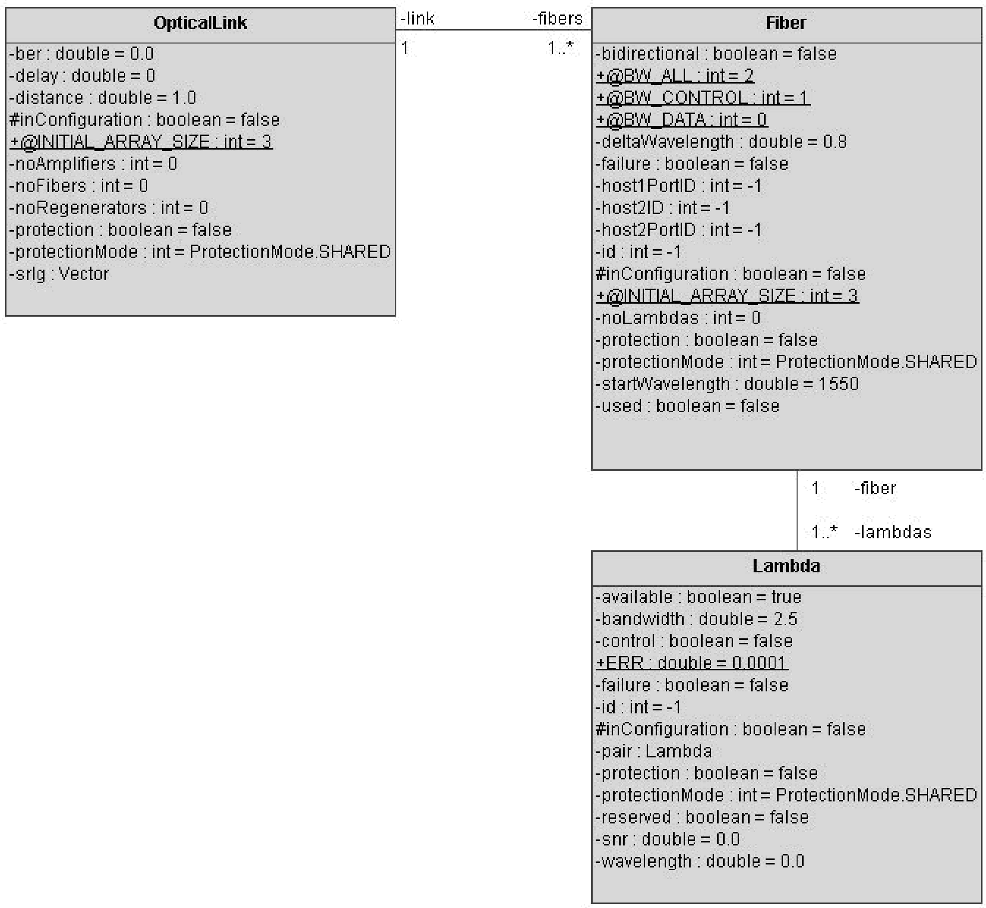


Figure 4: The Optical Link

In GLASS it is possible to monitor the traffic on the Link in both directions, outgoing and incoming. This is done by attaching monitors to the ONIC under consideration and reading the result with special players, which allow the post processing of the data stream.

2.2 THE LOGICAL COMPONENTS

The logical components reside on top of the physical layer topology. These components are used as interface between the simulator and the algorithms. The logical components are:

- Optical Route
- Optical Path
- Optical Channel
- Channel Segment

Each of these components has a variety of attributes and methods. The optical route contains the result of the routing algorithm. The optical path is the result of the wavelength assignment algorithm. The path contains multiple lambda channels (OpticalChannel). Each channel contains a list of segments and each segment is assigned to a corresponding lambda. This structure allows the routing and wavelength assignment algorithm to produce results without configuring the switches. This allows the implementer of algorithms as well as the implementer of protocols a flexible way of creating and establishing connections.

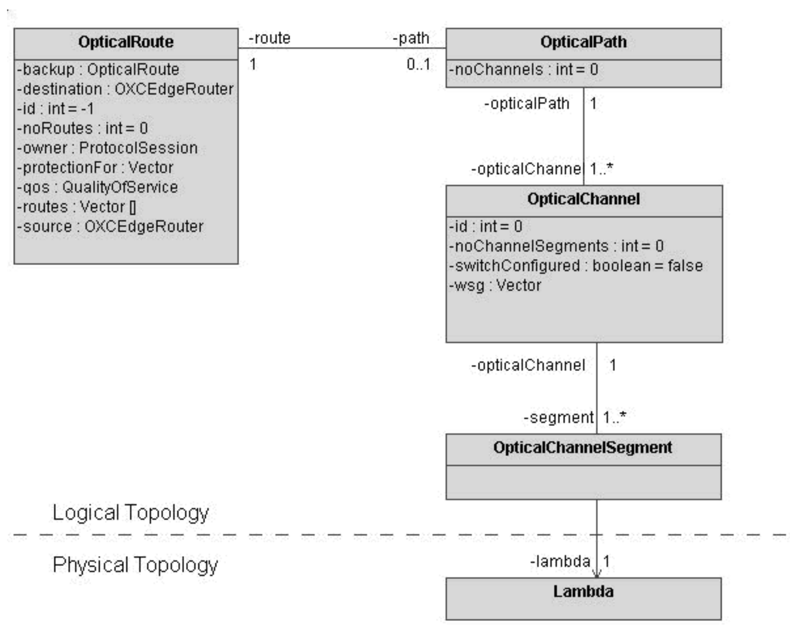


Figure 5: Logical Design

2.3 PROTOCOLS

GLASS provides access to entire library of the protocols available in SSF. These protocols run in the non-optical domain. At this time, on the optical side, GLASS provides the implementation of GMPLS, static OSPF over optical networks, IP over optical networks, diverse example implementations of failure propagation and recovery protocols that are not IP based (pure optical signaling).

2.4 THE SIMULATION

GLASS is a discrete event simulator. By default the GLASS framework provides failure and recovery events. GLASS allows protocols designers to use their own customized failure detection mechanisms. This has to be done by replacing the default modules for optical failure detection and non-optical failure detection. These modules are located in the network interface card (NIC) or the ONIC. Each card or connector has its own detection module. This allows users to run different detection strategies per node. The configuration of the failure detection modules will be done in the simulation DML file.

The simulator provides the following failures:

- Node Failure
- NIC / ONIC Failure
- Link / Optical Link Failure
- Fiber Failure
- Lambda Failure

Besides the failure handling, the simulator provides a couple of traffic generators.

3 THE VISUALIZATION TOOLS

There are two visualization tools provided in GLASS:

- GLASS – Browser
- GLASS – Topology and Simulation Creator (TSC)

3.1 THE GLASS BROWSER

The browser is a viewer on top of the simulation framework. It helps examine the actual state information of the physical and the logical topology. The browser allows running the simulation and examining the values of each component after the simulation ends.

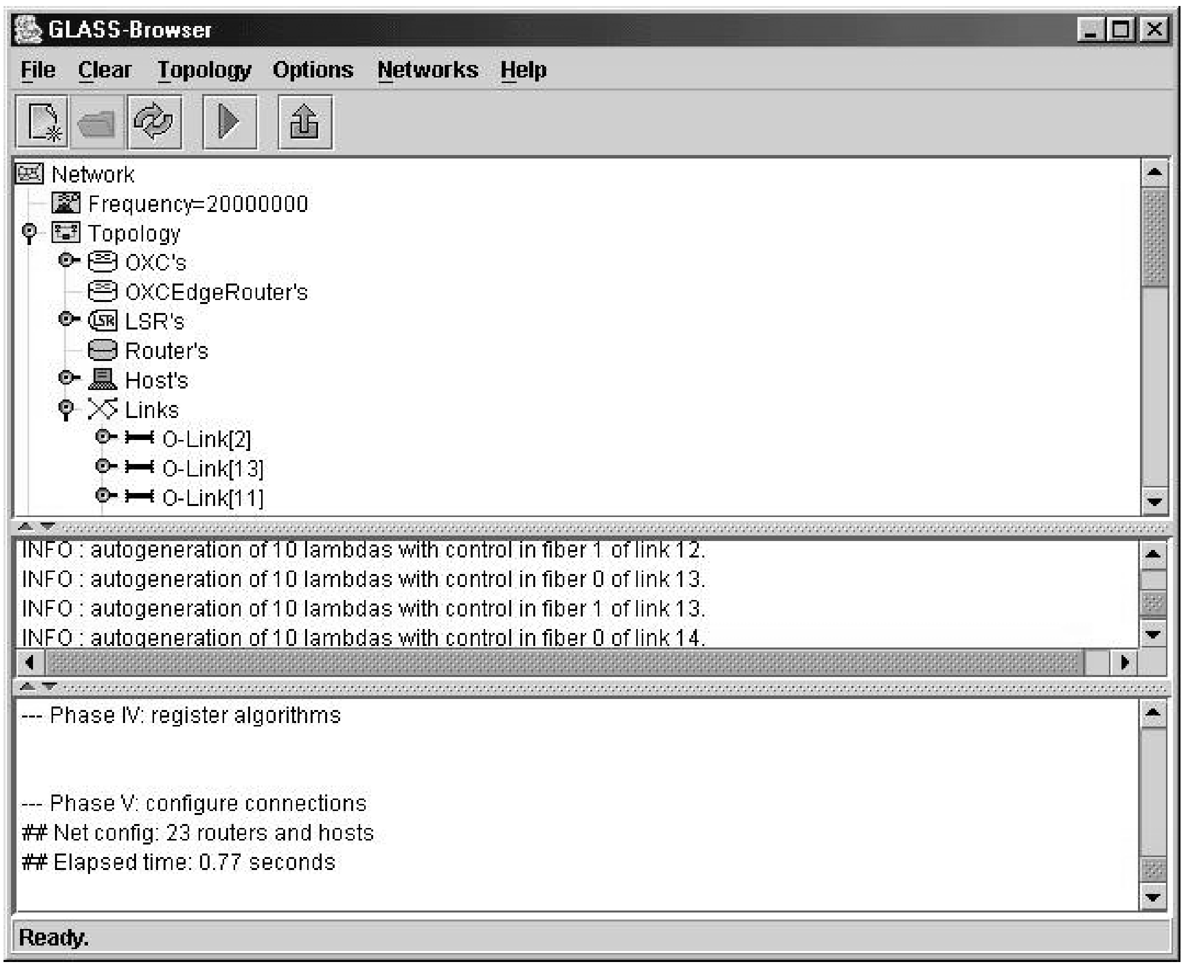


Figure 6: The GLASS Browser

3.2 THE GLASS-TSC

The GLASS-TSC allows the user to create a simulation by using a graphical user interface. Here the user can script failure and recovery events as well as configure the nodes, links, protocol stack, and predefine connections. The TSC also allows an algorithm developer to test his/her protocol without running the simulation. In this case, a user has to create the topology and specify connections between nodes.

The TSC also allows interactive simulations to improve debug capabilities.

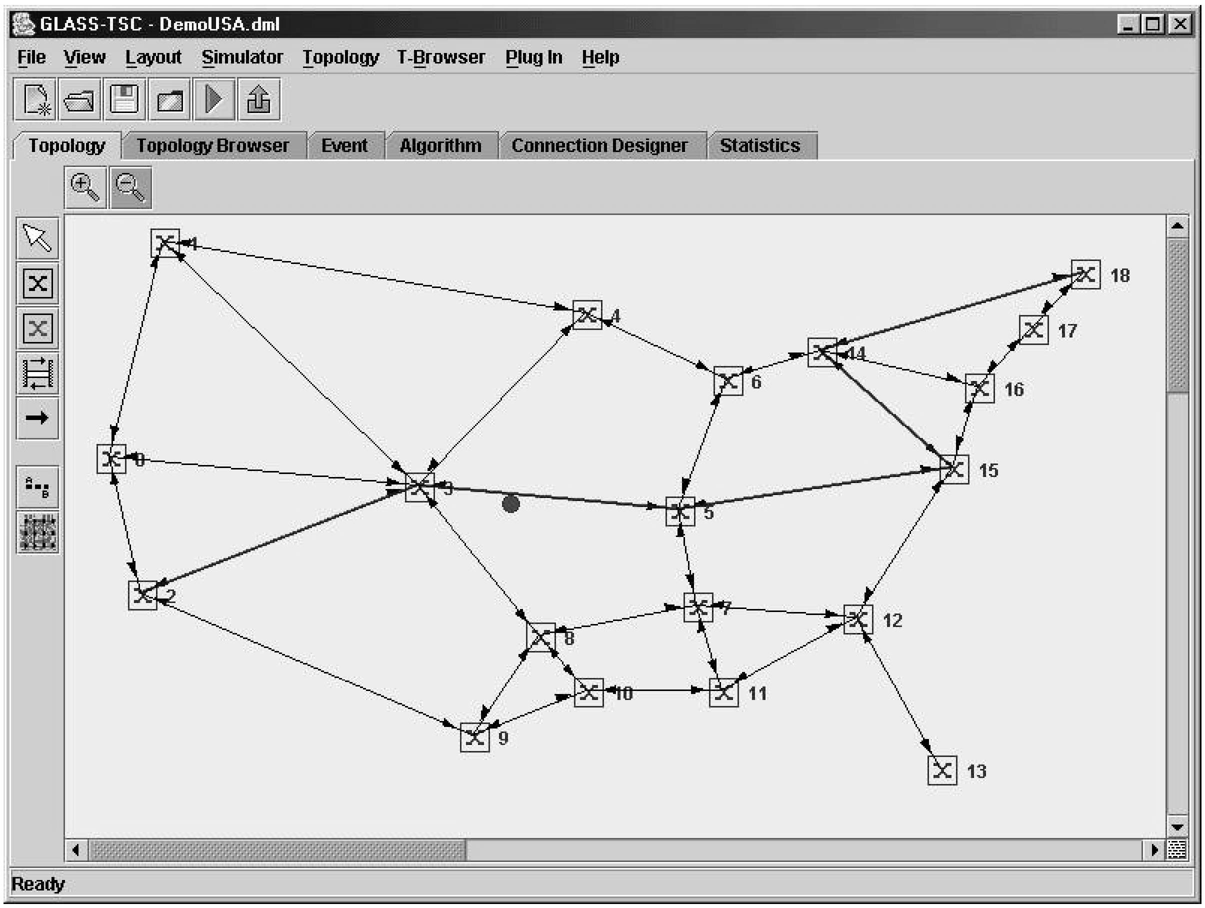


Figure 7: The GLASS TSC

REFERENCES

[1] Scalable Simulation Framework Network (SSFNet), http://www.ssfnet.org

[2] Data Modeling Language (DML), http://www.ssfnet.com/InternetDocs/ssfnetDMLReference.html

[3] Modeling Evaluation and Research of Lightwave Networks, MERLiN, NIST